

Spot Report: An Open-Source and Real-Time Secondary Task for Human-Robot Interaction User Experiments

Arsha Ali
Rohit Banerjee
Wonse Jo
University of Michigan
Ann Arbor, MI, USA

Kayla Riegner
Jonathon M. Smereka
Ground Vehicle Systems Center
Warren, MI, USA

Samantha Dubrow
MITRE Corporation
McLean, VA, USA

Lionel P. Robert Jr.
Dawn M. Tilbury
University of Michigan
Ann Arbor, MI, USA

ABSTRACT

The human-robot interaction (HRI) community is interested in a range of research questions, many of which are investigated through user experiments. Robots that occasionally require human input allow for humans to engage in secondary tasks. However, few secondary tasks transmit data in real-time and are openly available, which hinders interaction with the primary task and limits the ability of the community to build upon others' research. Also, the need for a secondary task relevant to the military was identified by subject matter experts. To address these concerns, this paper presents the spot report task as an open-source secondary task with real-time communication for use in HRI experiments. The spot report task requires counting target objects in static images. This paper includes details of the spot report task and real-time communication with a primary task. We developed the spot report task considering the military domain, but the software architecture is domain-independent. We hope others can leverage the spot report task in their own user experiments. The software files are available at <https://github.com/UMich-MAVRIC/SpotReport>.

CCS CONCEPTS

• **Human-centered computing** → **User studies**; *Laboratory experiments*; Interactive systems and tools.

DISTRIBUTION STATEMENT A. Approved for public release; distribution is unlimited. OPSEC #8254.

Approved for Public Release; Distribution Unlimited. Public Release Case Number 23-2193. The author's affiliation with The MITRE Corporation is provided for identification purposes only, and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions, or viewpoints expressed by the author.

The authors wish to acknowledge the technical and financial support of the Automotive Research Center (ARC) in accordance with Cooperative Agreement W56HZV-19-2-0001 U.S. Army DEVCOM Ground Vehicle Systems Center (GVSC) Warren, MI. Correspondence: arshaali@umich.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

HRI '24, March 11–14, 2024, Boulder, CO, USA

© 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 979-8-4007-0323-2/24/03...\$15.00

<https://doi.org/10.1145/3610978.3640718>

KEYWORDS

Human-Robot Interaction, Secondary Task, User Experiments, Research Tools, Open Source Software, Pygame, Lab Streaming Layer

ACM Reference Format:

Arsha Ali, Rohit Banerjee, Wonse Jo, Samantha Dubrow, Kayla Riegner, Jonathon M. Smereka, Lionel P. Robert Jr., and Dawn M. Tilbury. 2024. Spot Report: An Open-Source and Real-Time Secondary Task for Human-Robot Interaction User Experiments. In *Companion of the 2024 ACM/IEEE International Conference on Human-Robot Interaction (HRI '24 Companion)*, March 11–14, 2024, Boulder, CO, USA. ACM, New York, NY, USA, 5 pages. <https://doi.org/10.1145/3610978.3640718>

1 INTRODUCTION AND BACKGROUND

Humans and robots can work together, combining their strengths to handle tasks [1, 2]. In semi-autonomous systems, a robot can operate autonomously under certain conditions, but requires human assistance outside of these conditions [27]. As a result, semi-autonomous systems provide humans the opportunity to engage in a secondary task. A secondary task is a supplemental activity that is completed in conjunction with the primary task. Secondary tasks can be active or passive. Active secondary tasks require physical and/or mental operation, while passive secondary tasks do not [17]. Another way to classify secondary tasks is in terms of stimulus (e.g., visual, auditory) and response (e.g., manual, vocal) modalities [25]. In user experiments, a secondary task is designed to gather additional data or insights about the user's behavior, preferences, or performance [12]. For example, performance on a secondary task can provide a measure of spare capacity [10].

With semi-autonomous vehicles (AVs), secondary or non-driving related tasks (NDRTs) can be sufficiently engaging to distract or require dedicated attention away from the primary task of driving. Several NDRTs have been used in human-AV experiments, including watching a video [9, 15, 17], playing a tablet game [6, 16], playing a mobile phone game [13, 17], entering meaningless digits on a navigator screen [9], transcription [22, 24, 25], calendar entry [22], reading [18], verbally speaking [17, 24, 25], searching for the letter 'Q' in a field of 'O's [3–5], the standardized visual Surrogate Reference Task (SuRT) [7, 8, 19], the standardized cognitive n-back Task [19], and a twenty questions task [14] among others.

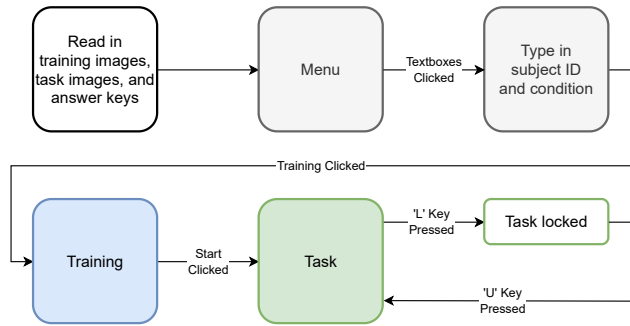


Figure 1: A summarized flowchart of the spot report program.

In the military, incorporating semi-AVs into human teams has the potential to decrease risks to human soldiers and enhance overall team capabilities [26]. Different from other domains, a secondary task in the military may be critical to the success of the primary task, thereby invoking different cognitive demands, engagement patterns, and performance. However, based upon discussions with military personnel and subject matter experts, there is a need to develop a realistic secondary task tailored to military operations. Moreover, the secondary task should be suitable and easy to integrate with various primary tasks. It is crucial that information and performance data from the secondary task are easily and continuously accessible in real-time, as they may impact delegation and performance on the primary task.

However, existing secondary tasks largely do not address these concerns. The lack of open-source secondary tasks also hinders the ability for researchers to adapt existing software to meet their needs. To tackle these concerns, we developed an open-source spot report secondary task for HRI user experiments. It is classified as an active or visual-manual task, where users count target objects in static images. The task is inspired by actual military spot reports where information that can affect future actions is shared and documented. Although the spot report task is motivated by military applications, it can be modified to suit other domains. The spot report task also enables real-time communication, allowing for integration with a variety of primary tasks. The goal of this paper is to present brief details about the spot report task and encourage other researchers to use the spot report task as appropriate to their work.

2 SPOT REPORT TASK DESCRIPTION

2.1 Software Program

The spot report program is developed in the Python language v3.9.7 [23] using the pygame library v2.4.0 [20]. The spot report program is summarized by the flowchart in Fig. 1. The three main screens of the spot report program are the menu, training, and task.

2.1.1 Menu. Pygame is initialized upon launching the spot report program, and the answer keys, training images, and task images are read. The menu (Fig. 2) is displayed using rectangular and text objects drawn on the screen. The pressing of the left mouse button over the subject ID or condition objects copies keyboard input onto the screen. Through this method, the subject ID and condition appear as text boxes and rectangular objects appear as clickable

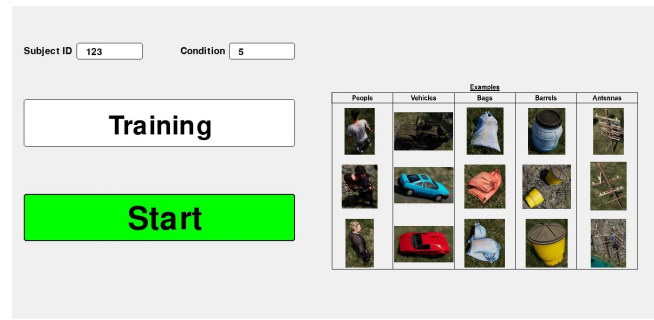


Figure 2: The spot report menu.

buttons. Pressing the escape key or clicking the exit button from the menu will close the entire spot report program.

2.1.2 Training. Once the subject ID and condition have been entered, the Training button can be clicked to enter the training loop. For training, the screen is set up with a training image to count objects in, buttons to increment and decrement the counts of the target object categories, a Next button, and labels for the current score and counts of each target object categories. The plus and minus buttons change the count of a target object category between 0 to 5. When the Next button is clicked, the score is updated and the next training image is shown. After the user has seen all 5 training images, the spot report program returns to the menu. Pressing the escape key or clicking the exit button during training will return the user to the menu.

2.1.3 Task. After completing training, the user can click the Start button to begin the experimental spot report task (Fig. 3). The spot report task is set up the same as in training, but the task images are used instead of the training images. Additionally, if the user has seen all 165 task images, the task images repeat from the beginning. When the Next button is clicked, the output files for accuracy, score, and task time are updated. Pressing the escape key or clicking the exit button during the task will close the entire spot report program. The task can be locked by pressing the 'L' key and then unlocked with the 'U' key. When locked, the user is prevented from seeing and interacting with the spot report task (Fig. 4). Locking the spot report task can force users to disengage from the secondary task and engage with the primary task instead.

2.2 Images

The training and task images were developed using Unreal Engine v5.0.3 [21]. There are five categories of objects that can be present in each image: people, vehicles, bags, barrels, and antennas. In each image, there are between 0 to 5 objects belonging to each category and a total of 0 to 10 objects. Across all of the task images, each object appears equally as often. Also, there are 15 task images with a total of i objects, $i \in \{0, \dots, 10\}$, for a total of 165 task images.

2.3 Point Scoring

In the current implementation, the performance of a user on the spot report task is determined through a point scoring system by comparing the counts submitted by the user to the values in the

Table 1: Points Earned Through Each Spot Report

Object	Points		Bonus
People	+2	× number of object	× correct count (0 or 1)
Vehicles	+1		
Bags	+1		
Barrels	+1		
Antennas	+1		
			+1

answer key. The answer key was created manually for each image. The point scoring system is summarized in Table 1. People are considered more important and are worth 2 points each compared to the other objects worth 1 point each. Note the user must submit the correct count for a target object category to earn points from that object category. If the user correctly counts all object categories for an image, 1 bonus point is awarded. The point scoring system can be adapted for other variations by modifying the `calculate_score` function in `score.py`. Point scoring can encourage users to complete the spot report task accurately and quickly in order to maximize their points.

2.4 Real-Time Communication

Lab Streaming Layer (LSL) is an open-source software framework to synchronize and stream data between different devices and softwares. LSL provides a standardized way of time-stamping and labeling data, making it easier to combine and analyze data from different sources [11]. We leverage LSL to facilitate real-time communication with a primary task. The LSL inlet and outlet streams are summarized in Table 2.

2.4.1 Inlet LSL Stream. The spot report task is controlled by an inlet stream called `spt_task_trigger`. This stream allows the user to lock or unlock the spot report task with a 0 or 1. When the inlet stream receives a 0, the task unlocks. When the inlet stream receives a 1, the task locks. This functionality represents how a primary task can control the spot report task, and is provided in addition to using the ‘L’ key to lock and the ‘U’ key to unlock the spot report task.

2.4.2 Outlet LSL Streams. The spot report task transmits data through various outlet streams.

Mouse information: The spot report task sends cursor positions and mouse button clicks during the experimental spot report task. Mouse information is transmitted through the `spt_mouse_pos` outlet stream for cursor positions and through the `spt_mouse_btn` outlet stream for mouse button clicks.

Task performance: Upon clicking the Next button, the execution time in seconds, accuracy, and score information are transmitted through the `spt_task_time`, `spt_task_accuracy`, and `spt_total_score` outlet streams respectively.

2.5 Output Files and Data

The output files `task_time.csv`, `accuracy.csv`, and `score.csv` log the execution time, accuracy, and score information, along with the current date and time and the image ID, every time the Next button is clicked. The execution time is the time spent on each task image, not including any time the spot report task is locked. The

Table 2: Lab Streaming Layer Outlet and Inlet Streams

Stream Type	Stream Name	Channel Details
Outlet 1	<code>spt_mouse_pos</code>	Ch 1: Image ID Ch 2: X position Ch 3: Y position
Outlet 2	<code>spt_mouse_btn</code>	Ch 1: Image ID Ch 2: 0 (for released) or 1 (for pressed)
Outlet 3	<code>spt_task_time</code>	Ch 1: Image ID Ch 2: Execution time (sec)
Outlet 4	<code>spt_task_accuracy</code>	Ch 1: Image ID Ch 2: Correct counts Ch 3: Incorrect counts Ch 4: Accuracy [%] Ch 5-9: counts of people, vehicles, bags, barrels, antennas
Outlet 5	<code>spt_total_score</code>	Ch 1: Image ID Ch 2: Image score Ch 3: Total score
Inlet 1	<code>spt_task_trigger</code>	0 (to unlock task) or 1 (to lock task)

score can also be logged by a change in points caused by a primary task, for example, if such information is sent on an LSL inlet stream. The output files `mouse_pos.csv` and `mouse_button.csv` log the mouse cursor position and the state of the mouse button (pressed or released) every time the values change. Such information, perhaps over a defined time interval, can provide additional real-time insight into the user’s cognitive states (e.g., engagement, spare capacity). Each of these output file names are appended with the subject ID and condition typed into the menu textboxes as `_S<subject ID>_C<condition>` to automatically save data from different users. Note that data is only logged from the experimental task and not from the menu or training.

3 ILLUSTRATIVE EXAMPLES

We conducted testing of the spot report program in Visual Studio Code. A video of the spot report program is available at <https://youtu.be/QspkXDuoXUM>. All software files are publicly available at <https://github.com/UMich-MAVRIC/SpotReport>. The program is executed from the terminal by entering `python .\spotreport.py`. This launches the spot report program and presents the menu, as depicted in Fig. 2. To enable the Training button, a subject ID and condition must be entered into the text boxes on the menu. After completing the training, the program returns to the menu. Training can be repeated if desired. After training, the Start button can be clicked to initiate the spot report task with the task images. When the spot report task is locked, a black screen with text is displayed as shown in Fig. 4.

Two examples of the spot report task with example counts that could be submitted by a user are shown in Fig. 3. The correct counts for the examples are listed under the “Number of Object” column along with the points earned in Table 3. In example A in Fig. 3a, all of the object categories are counted correctly and the 1 point bonus is earned. In example B Fig. 3b, the number of people and barrels are incorrectly counted so no points are earned from these categories and the bonus is not earned. Points are still earned from correctly counting the other object categories.



(a) An example where the user has counted all objects correctly.



(b) An example where the user has correctly counted the vehicles, bags, and antennas but incorrectly counted the people and barrels.

Figure 3: Examples of the spot report task.

Table 3: Points Earned in Example Task Images

Ex.	Object	Number of Object	Correct Count	Bonus	Points Earned
A	People	0	1	1	8
	Vehicles	3	1		
	Bags	1	1		
	Barrels	0	1		
	Antennas	3	1		
B	People	3	0	0	5
	Vehicles	0	1		
	Bags	4	1		
	Barrels	2	0		
	Antennas	1	1		

4 DISCUSSION

There is a wide range of secondary tasks being used in HRI user experiments. However, military personnel and subject matter experts have expressed the need for a secondary task that is relevant, engaging, and realistic in military settings. Furthermore, the secondary task should be adaptable to various primary tasks while continuously outputting data, which can be used in real-time by the primary task. The proposed spot report task addresses these requirements.

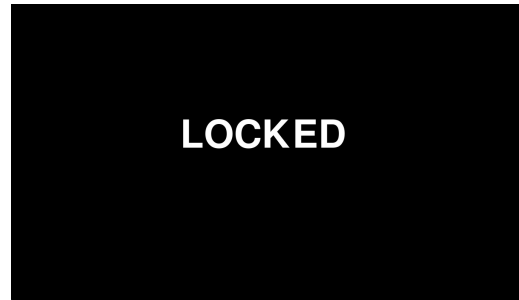


Figure 4: The display when the spot report task is locked.

By providing continuous and readily accessible data, the spot report task has the potential to enhance existing HRI user experiments and enable new ones. Many research topics in various domains can be pursued and impacted. For instance, the authors are using the spot report task in a user experiment to study team situational awareness and trust. In this experiment, a human subject collaborates with two semi-AVs to carry out the primary task of navigating and searching an off-road environment. When the human subject is not operating either of the semi-AVs, they are expected to engage in the spot report task which requires dedicated attention to earn points. The spot report task is presented as a case where images of the environment were taken by an unmanned aerial vehicle and identification of target objects can be used to determine the location of civilians. As such, performance on the spot report task can provide insight on the cognitive demands imposed by the primary task such as potential relationships with team situational awareness and trust. As another use case, real-time performance on the spot report task could be used to develop an adaptive primary task display with varying information and responsibilities to keep the user sufficiently engaged but not overloaded.

The spot report task can be leveraged in experiments where a secondary task would provide useful quantitative insights about human behavior. Empirical data and testing in various HRI scenarios is needed to demonstrate the effectiveness of the spot report task and its relationship to various cognitive constructs. Since the program is open-source, researchers have full freedom to modify and use any parts of the program as suitable for their purposes. For example, one can modify the spot report task to apply to another domain by using their own images, changing the point scoring system, or defining influence from a primary task through an LSL inlet stream. In addition, researchers could build upon the spot report task such as by scaling the complexity of images or imposing a time limit based on the user’s expertise or performance.

5 CONCLUSIONS

We presented the spot report task as a secondary task for use in human-robot interaction user experiments across domains. The spot report task is self-paced and entails counting target objects in images. The spot report task also outputs real-time data using LSL, which can be seamlessly integrated with a range of primary tasks. We hope making the program open-source and the details presented in this paper allow others to modify and adapt this work for their own purposes.

REFERENCES

- [1] Arsha Ali, Hebert Azevedo-Sa, Dawn M Tilbury, and Lionel P Robert Jr. 2022. Heterogeneous human-robot task allocation based on artificial trust. *Scientific Reports* 12, 15304 (2022), 1–15. <https://doi.org/10.1038/s41598-022-19140-5>
- [2] Arsha Ali, Dawn M. Tilbury, and Lionel P. Robert Jr. 2022. Considerations for Task Allocation in Human-Robot Teams. In *Proceedings of the AI-HRI Symposium at AAAI-FSS 2022*. <https://doi.org/10.7302/6543>
- [3] Hebert Azevedo-Sa, Suresh Kumar Jayaraman, Connor T Esterwood, X Jessie Yang, Lionel P Robert Jr., and Dawn M Tilbury. 2020. Comparing the Effects of False Alarms and Misses on Humans' Trust in (Semi) Autonomous Vehicles. In *Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction*. 113–115. <https://doi.org/10.1145/3371382.3378371>
- [4] Hebert Azevedo-Sa, Suresh Kumar Jayaraman, X Jessie Yang, Lionel P Robert Jr., and Dawn M Tilbury. 2020. Context-adaptive management of drivers' trust in automated vehicles. *IEEE Robotics and Automation Letters* 5, 4 (2020), 6908–6915. <https://doi.org/10.1109/lra.2020.3025736>
- [5] Hebert Azevedo-Sa, Huajing Zhao, Connor Esterwood, X Jessie Yang, Dawn M Tilbury, and Lionel P Robert Jr. 2021. How internal and external risks affect the relationships between trust and driver behavior in automated driving systems. *Transportation Research Part C: Emerging Technologies* 123 (2021), 1–17. <https://doi.org/10.1016/j.trc.2021.102973>
- [6] Sandra Epple, Fabienne Roche, and Stefan Brandenburg. 2018. The sooner the better: Drivers' reactions to two-step take-over requests in highly automated driving. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 62. 1883–1887. <https://doi.org/10.1177/1541931218621428>
- [7] Anna Feldhütter, Christian Gold, Sonja Schneider, and Klaus Bengler. 2017. How the duration of automated driving influences take-over performance and gaze behavior. In *Advances in Ergonomic Design of Systems, Products and Processes*. Springer, 309–318. https://doi.org/10.1007/978-3-662-53305-5_22
- [8] Christian Gold, Daniel Damböck, Lutz Lorenz, and Klaus Bengler. 2013. "Take over!" How long does it take to get the driver back into the loop?. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 57. 1938–1942. <https://doi.org/10.1177/1541931213571433>
- [9] Toshio Ito, Arata Takata, and Kenta Oosawa. 2016. Time Required for Take-over from Automated to Manual Driving. In *SAE 2016 World Congress and Exhibition*. <https://doi.org/10.4271/2016-01-0158>
- [10] Beth Kerr. 1973. Processing demands during mental operations. *Memory & Cognition* 1 (1973), 401–412. <https://doi.org/10.3758/bf03208899>
- [11] Christian Kothe, David Medine, Chadwick Boulay, Matthew Grivich, and Tristan Stenner. 2014. Lab Streaming Layer (LSL). <https://labstreaminglayer.org/#/>
- [12] Angela Mahr, Michael Feld, Mohammad Mehdi Moniri, and Rafael Math. 2012. The ConTRe (continuous tracking and reaction) task: A flexible approach for assessing driver cognitive workload with high sensitivity. In *Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*. 88–91. <https://www.auto-ui.org/12/adjunct-proceedings/w2-06-mahr.pdf>
- [13] Vivien Melcher, Stefan Rauh, Frederik Diederichs, Harald Widloither, and Wilhelm Bauer. 2015. Take-over requests for automated driving. *Procedia Manufacturing* 3 (2015), 2867–2873. <https://doi.org/10.1016/j.promfg.2015.07.788>
- [14] Natasha Merat, A Hamish Jamson, Frank CH Lai, and Oliver Carsten. 2012. Highly automated driving, secondary task performance, and driver state. *Human Factors* 54, 5 (2012), 762–771. <https://doi.org/10.1177/0018720812442087>
- [15] Brian Mok, Mishel Johns, Key Jung Lee, David Miller, David Sirkin, Page Ive, and Wendy Ju. 2015. Emergency, automation off: Unstructured transition timing for distracted drivers of automated vehicles. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2458–2464. <https://doi.org/10.1109/itsc.2015.396>
- [16] Brian Mok, Mishel Johns, David Miller, and Wendy Ju. 2017. Tunneled in: Drivers with active secondary tasks need more time to transition from automation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 2840–2844. <https://doi.org/10.1145/3025453.3025713>
- [17] Yutaka Nakajima and Kenji Tanaka. 2017. Effects of active and passive secondary tasks in a take-over situation during automated driving. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE, 1161–1166. <https://doi.org/10.1109/smc.2017.8122769>
- [18] Frederik Naujoks, Christoph Mai, and Alexandra Neukum. 2014. The effect of urgency of take-over requests during highly automated driving under distraction conditions. *Advances in Human Aspects of Transportation* 7, Part I (2014), 431. <https://doi.org/10.54941/ahfe100646>
- [19] Jonas Radlmayr, Christian Gold, Lutz Lorenz, Mehdi Farid, and Klaus Bengler. 2014. How traffic situations and non-driving related tasks affect the take-over quality in highly automated driving. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 58. 2063–2067. <https://doi.org/10.1177/1541931214581434>
- [20] A. Sweigart. 2012. *Making Games with Python & Pygame*. https://spada.uns.ac.id/pluginfile.php/621901/mod_resource/content/1/makinggames.pdf
- [21] Carlos Mauricio Torres-Ferreiros, Matthew Alexander Festini-Wendorff, and Pedro Nelson Shiguihara-Juárez. 2016. Developing a videogame using unreal engine based on a four stages methodology. IEEE, 1–4. <https://doi.org/10.1109/andescon.2016.7836249>
- [22] Remo MA van der Heiden, Shamsi T Iqbal, and Christian P Janssen. 2017. Priming drivers before handover in semi-autonomous cars. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 392–404. <https://doi.org/10.1145/3025453.3025507>
- [23] Guido Van Rossum and Fred L. Drake. 2009. *Python 3 Reference Manual*. CreateSpace.
- [24] Bernhard Wandtner, Gerald Schmidt, Nadja Schömig, and Wilfried Kunde. 2018. Non-driving related tasks in highly automated driving-effects of task modalities and cognitive workload on take-over performance. In *AmE 2018-Automotive meets Electronics; 9th GMM-Symposium*. VDE, 1–6. <https://ieeexplore.ieee.org/abstract/document/8402259>
- [25] Bernhard Wandtner, Nadja Schömig, and Gerald Schmidt. 2018. Effects of non-driving related task modalities on takeover performance in highly automated driving. *Human Factors* 60, 6 (2018), 870–881. <https://doi.org/10.1177/0018720818768199>
- [26] David P Watson and David H Scheidt. 2005. Autonomous systems. *Johns Hopkins APL Technical Digest* 26, 4 (2005), 368–376. <https://secwww.jhuapl.edu/techdigest/Content/techdigest/pdf/V26-N04/26-04-Watson.pdf>
- [27] Shlomo Zilberstein. 2015. Building Strong Semi-Autonomous Systems. *Proceedings of the AAAI Conference on Artificial Intelligence* 29 (2015). <https://doi.org/10.1609/aaai.v29i1.9773>